

BAB 2

DASAR TEORI

2.1 KAJIAN PUSTAKA

Penulisan laporan dan penelitian ini dilakukan dengan mengadaptasi dari beberapa jurnal yang berkaitan dengan rancang bangun sistem pemanas air yang menggunakan kendali PID. Seperti yang dilakukan oleh I.W. Parmadi Putra dan kawan-kawan pada tahun 2020 yang berjudul **“Perancangan Sistem Pemanas Air Menggunakan Sistem Kendali PID”**. Pada penelitian tersebut dilakukan perancangan sebuah sistem pemanas berbasis Arduino Uno dengan kendali PID. Metode tuning PID yang digunakan adalah trial and error yang mana dilakukan secara manual untuk memasukkan parameter Kp, Ki dan Kd sampai memperoleh respon sistem yang stabil. Alat utama yang digunakan yaitu Arduino Uno, Sensor suhu LM35 dan *Heater*. Pada penelitian ini penulis mengatur setpoint untuk menjaga suhu air sebesar 70°C dengan pengujian dilakukan sebanyak 3 kali yaitu menggunakan pengendali *Proportional* (P), *Proportional-Integral* (PI) dan *Proportional Integral Derivatif* (PID). Hasil menunjukkan bahwa dengan menggunakan pengendali PD respon sistem lebih stabil dibandingkan dengan pengendali P maupun PID. Namun pada pengendali PID menghasilkan nilai error yang kecil namun memerlukan waktu yang lebih besar untuk mencapai *setpoint*[6].

Penelitian oleh Andre Dharmawan dan kawan-kawan yang berjudul **“Sistem Kontrol Proporsional-Integral Pada Proses Pasteurisasi Susu”** pada tahun 2019. Penulis jurnal tersebut melakukan penelitian untuk membangun sistem berbasis mikrokontroler 68HC11A8 untuk Pasteurisasi susu yaitu proses pemanasan produk susu di bawah titik didih untuk membunuh bakteri untuk menjaga kualitas susu. Sensor yang digunakan pada penelitian tersebut adalah Sensor Temperatur MAX6675 *K-Thermocouple* untuk mengukur suhu pada produk susu. Metode pengujiannya menggunakan pengendali *Proportional* (P), *Proportional-Integral* (PI) dan *Proportional-Integral-Derivatif* (PID) dan *setpoint* diatur pada suhu 60°C. Hasil yang diperoleh adalah pengendalian suhu dengan metode *Proportional-Integral* (PI) menghasilkan respon yang paling baik dengan

keluaran suhu mencapai nilai maksimal 63°C . Penulis menyimpulkan bahwa sistem menggunakan pengendali PI menghasilkan respon yang paling baik dikarenakan keluaran atau respon suhu dari sistem hanya memiliki nilai *error* yang kecil dan waktu yang dibutuhkan sistem untuk mencapai *setpoint* yang cepat[7].

Berikutnya adalah penelitian yang dilakukan oleh Agus Riyanto dan M. Syafrullah dengan judul **“Pemantauan Suhu Pada Sistem Pemanas Air Menggunakan Temperatur Kontrol Dengan Metode PID Ziegler Nichols Berbasis Web”** pada tahun 2018. Penelitian ini penulis jurnal melakukan penelitian untuk merancang sebuah sistem yang dapat digunakan untuk pemantauan suhu pemanas air. Menggunakan pengendali PID dan menerapkan metode *Ziegler Nichols* untuk memperoleh nilai K_p , K_i dan K_d yang ideal agar menghasilkan respon sistem yang baik. Mikrokontroler yang digunakan yaitu berbasis esp8266 dengan sensor suhu PT100 dan pemanas air menggunakan alat pemanas air konvensional. Sebagai perbandingan, penulis juga melakukan pengujian tanpa pengendali PID. Hasil yang diperoleh pada sistem tanpa pengendali, dengan *setpoint* adalah 73°C dengan suhu terukur saat awal sistem dinyalakan adalah $36,69^{\circ}\text{C}$ memerlukan waktu 8 menit 14 detik untuk mencapai *setpoint*. Sedangkan pengujian dengan pengendali PID menggunakan *setpoint* yang sama dengan suhu awal air terukur sebesar $29,56^{\circ}\text{C}$ memerlukan waktu 19 menit 14 detik. Namun perbedaannya adalah dengan menggunakan pengendali PID sistem dapat mati apabila suhu air melebihi *setpoint* dan akan menyala apabila *setpoint* belum tercapai[8].

Pada penelitian yang dilakukan oleh Yulian Zetta Maulana dan Herryawan Pujiharsono dengan judul **“Perbandingan Kinerja Pengontrol PID menggunakan Antarmuka OPC pada PLC dan MATLAB untuk Sistem Pasteurisasi Susu”** pada tahun 2021, penulis membuat suatu *prototype* untuk pasteurisasi produk susu berbasis PLC dan MATLAB sebagai pengolah data. Penelitian tersebut menggunakan metode pengendali PI dan PID, sedangkan metode *tunning* untuk memperoleh parameter K_p , K_i dan K_d yang digunakan yaitu *Ziegler Nichols* dan *Cohen Coon*. Dengan mengatur *setpoint* yang digunakan adalah 72°C , penelitian yang dilakukan penulis menunjukkan hasil

yang paling stabil pada saat menggunakan pengendali sistem PID dan metode *tunning* yang digunakan adalah *Cohen Coon*. Hal tersebut dikarenakan respon sitem yang dihasilkan mampu mencapai *setpoint* dengan cepat yaitu dalam waktu 261 detik dan lebih stabil walaupun terdapat *error* sebesar 0,356 sehingga keluaran suhu sistem yaitu 70,66°C[9].

Berdasarkan penelitian-penelitian di atas, peneliti akan melakukan penelitian membangun sistem penghangat susu menggunakan pengendali PID dengan mengadaptasi metode tuning PID *Cohen Coon* dari penelitian pada referensi [9] yang dibuktikan dengan hasil data yang diperoleh dalam melakukan pasteurisasi suhu pada suhu *setpoint* 72°C mendapatkan hasil yang stabil dibandingkan dengan penelitian pada referensi[8]. Sedangkan perancangan alat *prototype* peneliti mengadaptasi dari penelitian pada referensi [6] yang mana menggunakan Arduino sebagai pusat kendali sistem, sensor suhu DHT11 sebagai pengukur suhu susu dan *Heater* sebagai pemanas.

Tabel 2. 1 Kajian Pustaka

Peneliti	Judul	Perbedaan	Hasil Penelitian
I.W. Parmadi Putra, ddk.	Perancangan Sistem Pemanas Air Menggunakan Sistem Kendali Pid (2020).	ASI, PTC <i>fan heater</i> , Oled display, metode <i>tunning Cohen Coon</i> .	Menghasilkan bahwa pengendali PD respon sistem lebih stabil dibandingkan dengan pengendali P maupun PID.
Andre Dharmawan, dkk.	Sistem Kontrol Proporsional-Integral Pada Proses Pasteurisasi Susu (2019).	ASI, Arduino UNO, sensor DHT11, metode <i>tunning Cohen Coon</i> .	Hasil menunjukkan bahwa pengendalian suhu dengan metode (PI) menghasilkan respon yang paling baik
Agus Riyanto, dan M. Syafrullah	Pemantauan Suhu Pada Sistem Pemanas Air Menggunakan Temperatur Kontrol Dengan Metode PID <i>Ziegler Nichols</i> Berbasis Web (2018).	ASI, Arduino UNO, sensor DHT11, metode <i>tunning Cohen Coon</i> .	Hasil pemantauan dari suhu awal terukur sampai pada nilai <i>setpoint</i> memerlukan waktu 8 menit 14 detik. Sedangkan Hasil pengujian dengan metode PID memerlukan waktu 19 menit 14 detik dan sistem dapat mati apabila suhu air melebihi <i>setpoint</i> dan akan menyala apabila <i>setpoint</i> belum tercapai.

Yulian Zetta Maulana, dan Herryawan Puji Harsono	Perbandingan Kinerja Pengontrol PID menggunakan Antarmuka OPC pada PLC dan MATLAB untuk Sistem Pasteurisasi Susu (2021).	ASI, Arduino UNO, sensor DHT11, Oled <i>display</i> .	Hasil yang paling stabil pada saat menggunakan pengendali sistem PID dan metode <i>tunning</i> yang digunakan adalah <i>Cohen Coon</i> . Hal tersebut dikarenakan respon sitem yang dihasilkan mampu mencapai <i>setpoint</i> dengan cepat.
--	--	---	---

2.2 DASAR TEORI

Pada bagian dasar teori ini menjelaskan mengenai teori-teori yang digunakan seperti teori metode yang digunakan sampai komponen yang digunakan. Dasar teori yang digunakan meliputi *bottle warmer*, karakteristik ASIP, pengendali PID, metode *tunning Cohen Coon*, mikrokontroler AT-mega 328 dan fitur yang digunakan, sensor DHT11, PTC *fan heater*, Oled *display*, dan module *driver* VNH2SP30. Berikut penjelasan mengenai teori-teori yang digunakan :

2.2.1 *Bottle Warmer*

Mengetahui cara memanaskan ASIP dapat membantu komposisi pasca penyimpanan karena saat ASIP dibekukan atau didinginkan, lemak cenderung terpisah dari botol. Menurut *Centers for Disease Control and Prevention (CDC)*, memanaskan ASIP dengan menggunakan *microwave* sangat dilarang karena gelombang mikro tidak memanaskan ASIP secara merata dan dapat membuat bayi kepanasan. Selain itu, diketahui bahwa gelombang mikro juga berpotensi merusak nutrisi dan antibodi yang terdapat pada kandungan ASIP[10].

Beberapa ibu menghangatkan ASIP dengan botol penghangat. Botol penghangat adalah alat yang mudah untuk memanaskan botol atau menghangatkan kantung ASIP. Namun, ada beberapa kelebihan dan kekurangan dalam penggunaannya. Salah satunya adalah efek negatif dari penghangat botol yaitu meningkatnya kemungkinan suhu ASIP menjadi terlalu panas, sehingga perlu pengawasan khusus. Maka dari itu perlu dikembangkannya sebuah *bottle warmer* yang dapat menjaga kestabilan suhu agar tidak terjadi suhu berlebih untuk menghindari hilangnya nutrisi pada ASIP[10].



Gambar 2. 1 *Bottle Warmer*[11]

2.2.2 Karakteristik ASIP

Air Susu Ibu (ASI) merupakan makanan pertama yang diberikan kepada bayi setelah lahir. Asi Eksklusif adalah ASI yang diberikan kepada bayi sejak dilahirkan sampai usia 6 bulan, tanpa menambahkan atau mengganti dengan nutrisi yang lain sebelum memasuki usia MPASI. Kandungan dalam ASI meliputi semua gizi yang dibutuhkan oleh bayi dalam usia 6 bulan seperti, lemak, karbohidrat, protein, vitamin, mineral, dan air. Selain itu ASI juga mengandung zat bioaktif yang membantu meningkatkan sistem kekebalan bayi. Semakin berkembangnya teknologi, ASI dapat diperah dan disimpan untuk memenuhi kebutuhan bayi ketika ditinggal oleh ibunya bekerja. Memompa ASI menjadi solusi untuk Ibu yang bekerja untuk tetap dapat memberikan ASI eksklusif kepada bayi. Setelah diperah ASIP sebaiknya disimpan dalam botol kaca atau kantung khusus ASIP. ASIP yang disimpan juga harus memperhatikan suhu tempat penyimpanan. ASIP yang baru diperah dalam suhu ruang dapat bertahan selama 24 jam dengan suhu 15⁰C, 10 jam dengan suhu 19-22⁰C, dan 4-6 jam dengan suhu 25⁰C. ASIP yang disimpan dalam Pendingin dapat bertahan 3-8 hari dengan suhu 0-4⁰C. Sedangkan ASIP yang disimpan dalam *freezer* dapat bertahan 3-4 bulan dalam *freezer* 2 pintu, dan 6-12 bulan dalam *freezer* khusus yang sangat dingin dengan suhu kurang dari 18⁰C. Kesalahan yang sering terjadi dalam manajemen ASIP adalah tidak memperhatikan metode penyimpanan dan memanaskan. ASIP

yang baru diperah tidak dapat langsung disimpan dalam *freezer*. ASIP yang telah dibekukan dalam *freezer* tidak dapat langsung dipanaskan dalam keadaan beku. ASIP yang telah dicairkan tidak boleh dibekukan kembali. Hal tersebut tidak diperbolehkan karena perubahan suhu ekstrim dapat merusak ASIP[12].

2.2.3 Sistem Kendali PID

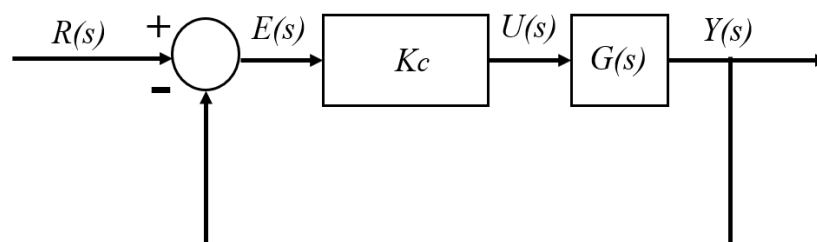
Kontrol atau kendali PID mencakup empat jenis pengendali yaitu Pengendali *Proportional* (P), Pengendali *Proportional* dan *Integral* (PI), Pengendali *Proportional Derivative* (PD) dan Pengontrol *Proportional Integral Derivative* (PID). Pada sub bab ini akan dibahas struktur dan parameter dari masing-masing kontroler PID. Berdasarkan pembahasan tersebut, maka akan mendapatkan informasi dasar tentang cara menggunakan pengendali PID. Berikut penjelasan mengenai pengendali :[13]

a. Pengendali *Proportional* (P)

Pengendali *Proportional* merupakan pengendali yang paling sederhana. Dengan istilah *proportional* ini, sinyal kontrol umpan balik $u(t)$ dihitung sebanding dengan kesalahan umpan balik $e(t)$ dengan rumus:

$$u(t) = K_c e(t) \quad (2.1)$$

Dimana K_c adalah penguatan *proportional* dan kesalahan umpan balik $e(t)$ adalah selisih antara sinyal referensi atau *setpoint* $r(t)$ dan sinyal *output* $y(t)$.



Gambar 2. 2 Konfigurasi Kontrol Umpan Balik Loop Tertutup[13]

Pada gambar di atas menunjukkan konfigurasi kontrol umpan balik *loop* tertutup di mana $R(s)$, $E(s)$, $U(s)$, dan $Y(s)$ adalah transformasi *Laplace* dari sinyal referensi atau *setpoint*, kesalahan umpan balik, sinyal kontrol, dan sinyal keluaran, masing-masing. $G(s)$ mewakili fungsi transfer *laplace* dari *plant*. Karena kesederhanaannya, pengontrol proporsional sering digunakan di mana sistem

memiliki sedikit informasi dan tidak memerlukan performa kestabilan yang baik. Karena pengendali hanya memiliki satu parameter untuk diatur, K_c dapat dipilih tanpa informasi *plant* yang detail[13].

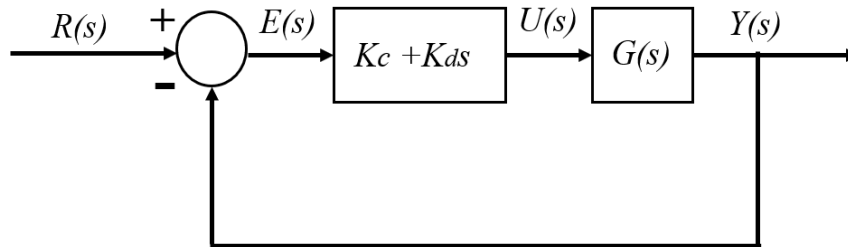
b. Pengendali *Proportional Derivative* (PD)

Dalam banyak aplikasi, kontroler proporsional K_c tidak cukup untuk mencapai tujuan kontrol tertentu seperti stabilisasi atau menghasilkan redaman osilasi yang memadai untuk sistem loop tertutup. Maka dari itu pengendali *Proportional Derivative* menjadi solusi untuk merancang sistem agar dapat meredam osilasi dan menghasilkan sinyal keluaran yang stabil. Pada pengontrol ini diperlukan variabel tambahan yaitu *derivative gain* (τ_D) yang dapat dinyatakan dalam rumus:

$$u(t) = K_c e(t) + K_c \tau_D \dot{e}(t) \quad (2.2)$$

Di mana dapat di transformasi menggunakan Laplace akan menjadi rumus seperti berikut:

$$\frac{U(s)}{E(s)} = K_c + K_c \tau_D s \quad (2.3)$$



Gambar 2. 3 Konfigurasi Umpan Balik Loop Tertutup Kontroler PD[13]

c. Pengendali *Proportional Integral* (PI)

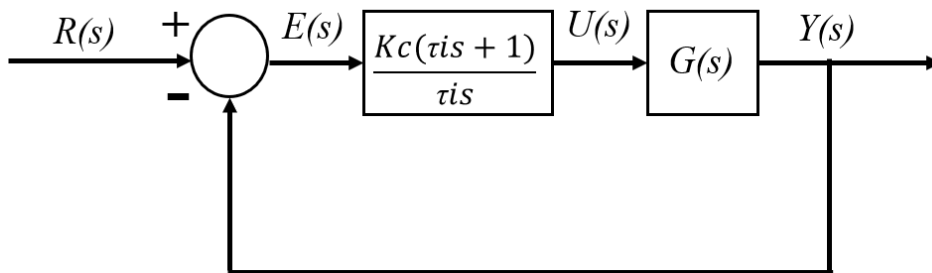
Pengendali *Proportional Integral* (PI) merupakan pengendali yang paling banyak digunakan di antara pengontrol PID. Dengan perhitungan integral, kesalahan *steady state* yang ada dengan kontroler proporsional akan benar-benar dihilangkan. *Output* dari pengontrol $u(t)$ adalah jumlah dari dua suku, satu dari fungsi proporsional dan yang lainnya adalah aksi integral, sehingga dirumuskan sebagai berikut:

$$u(t) = K_c e(t) + \frac{K_c}{\tau_I} \int_0^t e(\tau) d\tau \quad (2.4)$$

Di mana $e(t) = r(t) - y(t)$ adalah sinyal *error* antara sinyal *setpoint* $r(t)$ dan *output* $y(t)$, K adalah penguatan proporsional, dan τ_I adalah konstanta waktu integral. Parameter τ_I selalu positif, dan nilainya berbanding terbalik dengan pengaruh aksi integral yang diambil oleh kontroler PI. τ_I yang lebih kecil akan menghasilkan efek aksi integral yang besar.

Dari persamaan 1.4 apabila ditransformasi *laplace* akan menjadi rumus berikut:

$$\frac{U(s)}{E(s)} = \frac{Kc(\tau_I s + 1)}{\tau_I s} \quad (2.5)$$



Gambar 2. 4 Konfigurasi Umpan Balik Loop Tertutup Kontroler PI[13]

d. Pengendali *Proportional Integral Derivative* (PID)

Pengendali PID terdiri dari tiga suku: suku proporsional (P), suku integral(I), dan suku turunan (D). Dalam bentuk yang ideal, keluaran $u(t)$ dari pengontrol PID adalah jumlah dari tiga suku:

$$u(t) = Kc e(t) + \frac{Kc}{\tau_I} \int_0^t e(\tau) d\tau + Kc\tau_D \frac{de(t)}{dt} \quad (2.6)$$

Di mana $e(t) = r(t) - y(t)$ adalah sinyal kesalahan umpan balik antara sinyal referensir(t) dan keluaran $y(t)$, dan τ_D adalah gain kontrol turunan. Fungsi *Laplace* transfer dari pengontrol PID adalah:[14]

$$\frac{U(s)}{E(s)} = Kc \left(1 + \frac{1}{\tau_I s} + \tau_D s \right) \quad \text{pers (2.7)}$$

2.2.4 *Tunning Method*

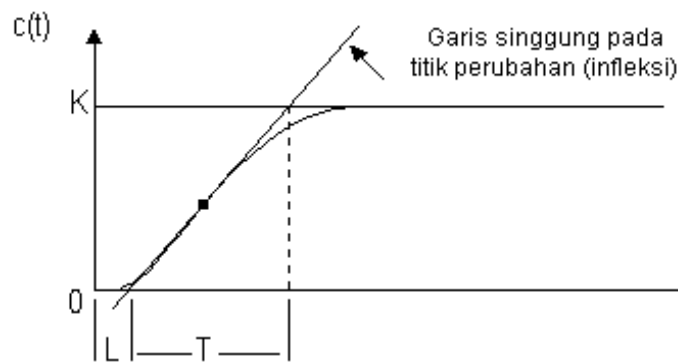
Tunning kontroler adalah proses penentuan parameter kontroler yang akan menghasilkan *output* yang diinginkan. Saat ini, lebih dari separuh kontroler yang digunakan di industri adalah kontroler PID. Sebelumnya kebanyakan jenis pengontrol yang digunakan adalah pengontrol analog. Namun saat ini, pengontrol

yang digunakan berbentuk sinyal digital dan komputer. Setelah model matematis sistem tersedia, parameter kontrol dapat ditentukan secara eksplisit. Namun, jika model matematis tidak tersedia, parameter harus ditentukan secara eksperimental. Pencocokan pengontrol memungkinkan pengoptimalan proses dan meminimalkan kesalahan antara variabel proses dan titik setelnya[15].

Metode *tunning* pengontrol yang paling sering digunakan adalah metode *Ziegler-Nichols* dan *Cohen-Coon*. Metode ini sering digunakan ketika model matematis dari sistem tidak tersedia. Metode *Ziegler-Nichols* dapat digunakan pada sistem *loop* tertutup dan *loop* terbuka, sedangkan metode *Cohen-Coon* umumnya digunakan pada sistem *loop* terbuka[15].

2.2.4.1 Ziegler Nichols Tuning Method

Metode *Ziegler Nichols* memiliki 2 metode *tunning* yaitu metode metode osilasi dan kurva reaksi. Pada metode osilasi biasa digunakan untuk sistem *loop* tertutup. Sedangkan metode kurva reaksi digunakan untuk sistem *loop* terbuka. Metode kurva reaksi memiliki kurva berbentuk S dan memiliki 2 konstanta yaitu T (waktu tunda) dan L (waktu mati) seperti pada grafik dibawah ini [16] :



Gambar 2. 5 Tanggapan Kurva S[16]

Pada gambar 2.5 menjelaskan kurva reaksi berubah naik setelah adanya selang waktu mati (L) dan waktu tunda (T) menjelaskan perubahan kurva setelah 66% dan berada pada *setpoint*. Berikut tabel perhitungan *tunning Ziegler Nichols* dengan menggunakan metode kurva reaksi :

Tabel 2. 2 Aturan *Tunning Ziegler-Nichols tipe-1* [16]

Tipe Kontroler	Kp	Ti	Td	Ki	Kd
P	T/L	∞	0		
PI	0,9 T/L	L/0,3	0		
PID	1,2 T/L	2L	0,5L	Kp/Ti	Kp.Td

2.2.4.2 *Cohen Coon Tunning Method*

Metode *tunning Cohen-Coon* diklasifikasikan sebagai metode '*offline tuning*', yang artinya perubahan *step* dapat dimasukkan ke *input* setelah sistem berada pada kondisi stabil. Kemudian *output* dapat diukur berdasarkan konstanta waktu dan waktu tunda dan respon ini dapat digunakan untuk mengevaluasi parameter kontrol awal.

Metode *tunning Cohen-Coon* digunakan untuk mengkoreksi respon yang lambat, dan kondisi *steady state* yang diberikan oleh metode *Ziegler-Nichols* ketika ada waktu mati (penundaan proses) terhadap konstanta waktu *loop* terbuka. Metode ini hanya digunakan untuk model orde pertama dengan waktu tunda, karena pengontrol tidak menanggapi gangguan secara instan. Metode *Cohen Coon* dapat dihitung menggunakan parameter *plant* yang didapat dari kurva reaksi seperti pada tabel berikut :[17]

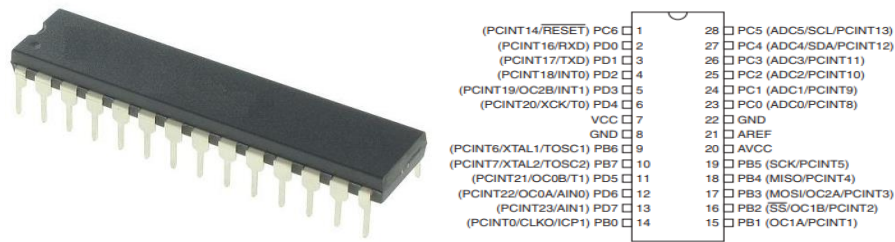
Tabel 2. 3 Penalaan Paramater PID dengan Metode *Cohen-Coon* [17]

Tipe kontroler	Kp	Ti	Td
P	$\frac{1}{K} \left(\frac{T}{L} \right) \left[1 + \frac{1}{3} \frac{L}{T} \right]$	-	-
PI	$\frac{1}{K} \left(\frac{T}{L} \right) \left[0,9 + \frac{1}{12} \frac{T}{L} \right]$	$L \left[\frac{30 + 3 \frac{L}{T}}{9 + 20 \frac{L}{T}} \right]$	-
PD	$\frac{1}{K} \left(\frac{T}{L} \right) \left[\frac{5}{4} + \frac{1}{6} \frac{L}{T} \right]$		$L \left[\frac{6 - 2 \frac{L}{T}}{22 + 3 \frac{L}{T}} \right]$
PID	$\frac{1}{K} \left(\frac{T}{L} \right) \left[\frac{4}{3} + \frac{1}{4} \frac{L}{T} \right]$	$L \left[\frac{32 + 6 \frac{L}{T}}{13 + 8 \frac{L}{T}} \right]$	$L \left[\frac{4}{11 + 2 \frac{L}{T}} \right]$

2.2.5 AT-Mega328

ATmega328 merupakan *chip* mikrokontroler CMOS 8-bit berdaya rendah (*low-power*) berbasis arsitektur AVR-RISC. ATmega328 memiliki fitur-fitur seperti 32K *byte flash* yang dapat diprogram dalam sistem dengan kemampuan *read-write*, EEPROM 1K *byte*, SRAM 2K *byte*, 23 jalur I/O GPIO, 32 register, 3 *Timer/Counter*, interupsi internal dan eksternal, USART serial yang dapat diprogram, antarmuka serial 2 kabel berorientasi *byte*, port serial SPI, ADC 10 bit 6 *channel* (8 *channel* dalam TQFP dan QFN/MLF paket), pengatur waktu yang dapat diprogram dengan osilator internal, dan lima mode hemat daya sebagai berikut :[18]

- a. Mode diam, yang berfungsi untuk menghentikan CPU sambil mengizinkan SRAM, *Timer/Counter*, USART, antarmuka serial 2-kawat, *port* SPI, dan sistem interupsi untuk terus berfungsi.
- b. Mode *power-down*, yang berfungsi untuk menyimpan isi register tetapi membekukan (*freezes*) osilator, menonaktifkan semua *chip* lainnya berfungsi hingga interupsi berikutnya atau reset perangkat keras.
- c. Mode hemat daya, yang berfungsi untuk pengatur waktu asinkron yang terus berjalan, dan memungkinkan pengguna untuk mempertahankan basis pengatur waktu saat perangkat lainnya sedang tidur.
- d. Mode *noise reduction*, yang berfungsi saat ADC menghentikan CPU dan semua modul I/O kecuali timer asinkron dan ADC, untuk meminimalkan derau atau *noise switching* selama konversi ADC. Dalam keadaan siaga
- e. Mode *Standby*, yang berfungsi untuk osilator kristal atau resonator berjalan saat perangkat lainnya sedang non aktif. Hal ini memungkinkan *start-up* yang sangat cepat dikombinasikan dengan konsumsi daya yang rendah[18].



Gambar 2. 6 IC AT-Mega328 dan Pin Out AT-Mega328[18]

2.2.5.1 ADC AT-Mega328

Ketika menghubungkan sensor ke mikrokontroler, *output* dari sensor seringnya bersifat analog. Tetapi mikrokontroler memproses sinyal digital. Maka, memerlukan ADC pada mikrokontroler untuk mengubah sinyal analog menjadi sinyal digital agar dapat diolah mikrokontroler AT-Mega328. Ada banyak aplikasi ADC seperti aplikasi biometrik, Pemantauan lingkungan, Deteksi kebocoran gas, dll[19].

AT-Mega328 memiliki 6 channel ADC *on-board* yang dapat digunakan untuk membaca sinyal analog dengan range 0-5V. ADC pada AT-Mega328 memiliki 10-bit ADC yang artinya akan memberikan nilai digital dalam kisaran 0 – 1023 (2^{10}) yang linear dengan *range* masukan analog 0-5V. Nilai 10-bit disebut sebagai resolusi yang menunjukkan jumlah nilai diskrit yang dapat dihasilkannya pada rentang nilai analog. Pin ADC pada AT-Mega328 antara lain PC0-PC5 atau pin A0-A5 pada *board* Arduino Uno[19].

2.2.5.2 Pin I/O AT-Mega328

General-Purpose Input Output (GPIO) adalah pin digital dari sebuah IC AT-Mega 328 yang terdiri dari pin PD0-PD7 atau jika di board Arduino Uno didefinisikan pin GPIO 0-7, dan PB0-PB5 atau pin GPIO 8-13 jika di board Arduino Uno. Pin ini digunakan sebagai *input* atau *output* untuk perangkat seperti sensor maupun aktuator.

Jika ingin membaca status saklar, data sensor, dll, maka mengkonfigurasinya sebagai *input*. Dan jika ingin mengontrol lampu LED, putaran motor, menampilkan teks pada layar, dll, maka mengkonfigurasinya sebagai *output*[18].

2.2.5.3 Komunikasi I2C AT-Mega328

I2C (*Inter-Integrated Circuit*) adalah protokol koneksi antarmuka bus serial. Komunikasi ini juga disebut TWI (*Two Wire Interface*) karena hanya menggunakan dua kabel untuk komunikasi yaitu SDA (serial data) dan SCL (serial clock).

I2C merupakan protokol komunikasi berdasarkan konfirmasi, pengirim mengkonfirmasi kepada penerima setelah mengirim data untuk menentukan apakah data telah berhasil diterima oleh penerima. I2C bekerja dalam dua mode, mode *master* dan mode *slave*. SDA (Serial Data) digunakan untuk bertukar data antara perangkat *master* dan *slave*. SCL (Serial Clock) digunakan untuk sinkronisasi antara perangkat *master* dan *slave*.

Perangkat master yang memulai komunikasi dengan perangkat *slave* memerlukan alamat perangkat *slave* untuk memulai komunikasi. Perangkat *slave* merespons perangkat *master* saat dialamatkan oleh perangkat *master*. Perangkat I2C memiliki alamat unik 7-bit atau 10-bit. Jadi, untuk mengakses perangkat ini, master harus mengalamatkannya dengan alamat unik 7-bit atau 10-bit.

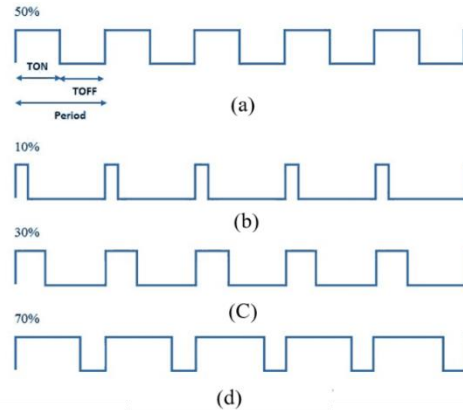
I2C digunakan di banyak aplikasi seperti membaca RTC (*Real-time clock*), mengakses memori EEPROM eksternal. I2C juga digunakan dalam modul sensor. Protokol I2C menggunakan 2 jalur untuk komunikasi yaitu *Serial Clock* (SCL) yang mana data akan dikirim ke perangkat lain berdasarkan *clock tick event*. Hanya perangkat master yang memiliki kendali atas jalur SCL ini. Kemudian *Serial Data* (SDA) yang mana jalur data serial yang digunakan untuk bertukar data antara perangkat *master* dan *slave*[20].

2.2.5.4 PWM AT-Mega328

Pulse Width Modulation (PWM) merupakan teknik yang memvariasikan lebar pulsa dengan menjaga frekuensi bentuk gelombang konstan. Sinyal PWM terdiri dari dua komponen utama yang menentukan perilakunya: *Duty Cycle* dan frekuensi. *Duty Cycle* merupakan periode pulsa yang terdiri dari siklus *ON* (5V) dan siklus *OFF* (0V) yang dapat dirumuskan dengan persamaan berikut:

$$Duty\ Cycle = \frac{T_{on}}{T_{on}+T_{off}} \times 100\% \quad (2.8)$$

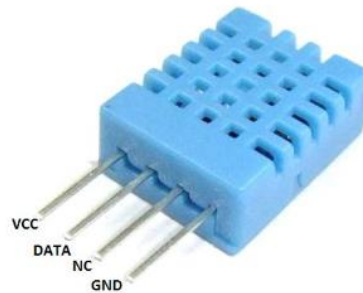
Dengan menggunakan teknik PWM, pengguna dapat mengontrol daya yang dikirimkan ke beban dengan menggunakan sinyal *ON-OFF*. Sinyal PWM dapat digunakan untuk mengontrol kecepatan motor DC dan mengubah intensitas LED. Pada AT-Mega pin PWM yang digunakan adalah pin PD3, PD5, PD6, PB1-PB3 atau pada *board* Arduino Uno adalah pin 3, 5, 6 dan 9-11[21].



Gambar 2. 7 Sinyal Duty Cycle, (a) 50% Duty Cycle, (b) 10% Duty Cycle, (c) 30% Duty Cycle, (d) 70% Duty Cycle[21]

2.2.6 Sensor DHT11

Sensor DHT11 merupakan modul sensor yang bekerja untuk mendeteksi suhu dan kelembaban pada suatu objek yang memiliki *output* berupa sinyal digital. Sensor ini memiliki jangkauan pengukuran suhu sebesar 0-50°C dan jangkauan pengukuran kelembaban relatif sebesar 20-90%. Sensor ini membutuhkan *power supply* sebesar 3-5,5 Volt DC. Dalam sensor ini terdapat sebuah thermistor dengan tipe NTC (*Negative Temperature Coefficient*) sehingga untuk cara kerja pengukuran suhu pada sensor ini yaitu semakin tinggi suhu di sekitar sensor maka nilai resistansi NTC akan semakin kecil. Sebaliknya nilai resistansi-nya akan meningkat ketika suhu ruangan sensor menurun[22].



Gambar 2. 8 Sensor DHT11[22]

2.2.7 DC PTC Fan Heater

Heater merupakan sebuah alat yang mampu mengubah energi listrik menjadi energi panas. Dalam kehidupan sehari-hari *heater* sering dipakai pada peralatan seperti setrika, *hair dryer*, pemanas air, pemasak nasi, oven dan sebagainya. Setiap pemanas terdapat elemen yang difungsikan sebagai penghasil panas yang biasanya berisi lilitan atau *coil* untuk memanaskan elemen tersebut. Elemen pemanas bisa berbentuk tabung ataupun plat tergantung alat dan pengaplikasiannya, contohnya setrika yang menggunakan elemen pemanas berbentuk tabung.

Kebanyakan alat rumah tangga yang menggunakan pemanas pada jaman dulu masih dioperasikan semi otomatis. Dimana alat tersebut bekerja dengan cara elemen akan aktif ketika suhu belum mencapai nilai yang dijadikan acuan atau *setpoint*, dan akan mati apabila suhu melebihi *setpoint*. Hal tersebut menyebabkan ketidak-stabilnya suhu yang dihasilkan terhadap *setpoint* yang diharapkan.

Saat ini teknologi elemen pemanas semakin berkembang agar dapat menghasilkan suhu yang stabil, hemat energi dan praktis salah satunya ialah elemen PTC atau *Positive Temperature Coefficient*. PTC merupakan komponen semikonduktor yang mampu menghasilkan koefisien suhu positif yang besar. Jika suatu bahan menunjukkan koefisien suhu positif berarti ketika suhunya meningkat, nilai resistansi terhadap arus meningkat secara proporsional. Awalnya, arus yang melalui PTC ini akan meningkat tajam dengan resistansi rendah tetapi kemudian turun menjadi nol saat suhu *setpoint* tercapai.

Keunggulan PTC dibandingkan dengan elemen pemanas berbahan dasar kawat adalah elemen pemanas PTC dibuat menggunakan bahan batu keramik yang diolah dengan barium titanate. Batu keramik ini memiliki karakteristik yang sangat unik sebagai semikonduktor yang sifat-sifatnya dapat ditentukan secara tepat selama desain dan pembuatannya. Perangkat PTC tidak membakar oksigen sehingga lebih aman digunakan di lingkungan yang terkendali. Hal ini juga membuat komponen tersebut lebih hemat energi. Pemanas PTC tidak bisa *overheat* dimana faktor unik ini membuat perangkat membatasi diri ketika resistansinya meningkat, arus melalui komponen berkurang. Mereka juga dapat dianggap mampu mengendalikan diri. Jika suhu sekitar mulai menurun, resistansi

pemanas akan berkurang menarik lebih banyak arus sehingga suhu yang dikeluarkan akan stabil dan sebaliknya dan karakteristik ini tidak terdapat pada pemanas konvensional[23].



Gambar 2. 9 DC PTC Fan Heater[24]

2.2.8 Oled Display

OLED *display* adalah panel visual elektronik yang memanfaatkan *Light Emitting Diode* (LED) berbahan organik untuk penerangan intinya. OLED adalah jenis teknologi layar *electroluminescent*, di mana lapisan bahan organik menghasilkan cahaya ketika molekul dioda teraliri oleh arus listrik. OLED dapat digunakan untuk menampilkan gambar, teks, video, dan lainnya di layar atau panel dengan hampir semua ukuran. Berkat keunikannya yang diberikan dalam hal daya dan kinerja, layar OLED juga semakin banyak digunakan sebagai alat tampilan kinerja di semua industri dan sektor saat ini.

Pada proses *prototyping*, jenis OLED berukuran 0,91” dengan ukuran 128x32 *pixel* merupakan jenis OLED yang sering digunakan. Hal itu dikarenakan fitur komunikasi I2C yang tertanam pada komponen tersebut yang memudahkan pengguna dalam melakukan pemrograman ataupun instalasi. OLED ini dapat diimplementasikan ke berbagai mikrokontroler seperti Arduino, STM32, ESP8266 maupun Raspberry Pi[24].



Gambar 2. 9 OLED 0,91” 128x32 Pixel[24]

2.2.9 Driver Motor Dc VNH2SP30

Analog to Digital Converter (ADC) digunakan pada mikrokontroler untuk mengubah tegangan input yang berupa nilai analog menjadi nilai digital. Namun hal ini berbeda dengan konversi digital ke analog atau *Digital to Analog* (DAC). Sebagai gantinya digunakanlah metode modulasi lebar pulsa atau *Pulse Width Modulated* (PWM) untuk menghasilkan *output* analog. Penerapan PWM seperti pada kontrol motor, membuat tegangan linier atau *driver* arus yang memerlukan DAC murni. Untuk membuat DAC murni dapat dilakukan dengan cara mengubah *output* PWM ke level tegangan analog. Dalam perancangan DAC murni memerlukan *low-pass* filter sederhana yang terbuat dari resistor dan kapasitor keramik. *Low-pass* filter RC sederhana mengubah sinyal PWM menjadi tegangan yang sebanding dengan *duty cycle* Namun pada penggunaan *low-pass filter* sederhana hanya dapat bekerja pada *output* tegangan dari mikrokontroler yang digunakan. Contohnya pada Arduino yang mengeluarkan tegangan 5V, sehingga jika diubah ke dalam bentuk sinyal analog maka tegangan yang keluar juga 5V.

Hal tersebut tidak dapat digunakan untuk men-*drive* sebuah aktuator yang membutuhkan tegangan lebih dari 5V. Apabila ingin mengubah PWM ke tegangan yang lebih tinggi maka dapat menggunakan modul *driver*. Terdapat beberapa varian modul *driver*, salah satunya adalah *driver* motor DC VNH2SP30. *Driver* motor DC VNH2SP30 berfungsi sebagai pengontrol motor DC dengan tegangan dan arus yang tinggi. *Driver* motor DC VNH2SP30 memiliki keluaran yang dapat digunakan sebagai pengendali motor DC yang memiliki tegangan maksimal sebesar 16 V dan arus maksimal sebesar 30 A[25].



Gambar 2. 10 *Driver* motor DC VNH2SP30[25]